Осетин Михаил Андреевич.

Организация ЭВМ и систем

Литература:

1) Орлов С.А., Б.Я. Цилькер.
"Организация ЭВМ и систем", 2011, 688стр.
2) Горнец Н.Н., Рощин А.Г, Соломенцев А.В
"Организация ЭВМ и систем", 2008, 320стр.
3) Бродин В.Е, Шнурин И.И
"Микропроцессор Intel 486
архитектура, программирование, интерфейс,"
1993, 240с

1) Архитектура процессора
2) Система процессор-память
3) Защищенный режим работы проц.
4) Организация совершенных ЭВМ
5) Микропроц. вычисл. системы

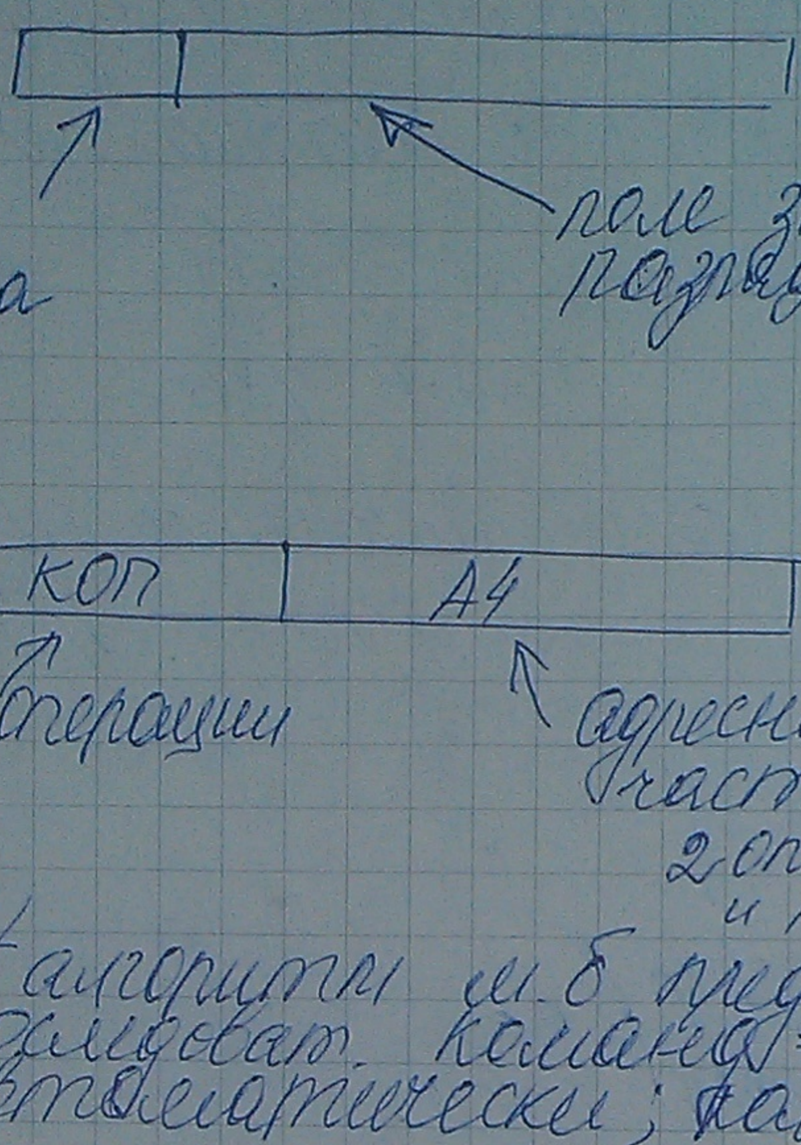
Лекция №1

Концепция выч. машины была предложена фон Неймановой в 1945г.

Сущность концепции:

1) Двоичное кодирование
2) Прогр. управление
3) Однородность памяти
4) Адресуемость памяти

1) Вся инф. в 2-м коде: данные и команды. Они загруж. в память ВМ и вводе неотличимы, можно только извлекая из памяти. Опр. формат данных и команд: байт, слово, двойное слово. Формат предусматр. поля.



2) Алгоритм м.б представлен в виде прогр., полностью команда ⇒ машина работает автоматически; команды друг за другом

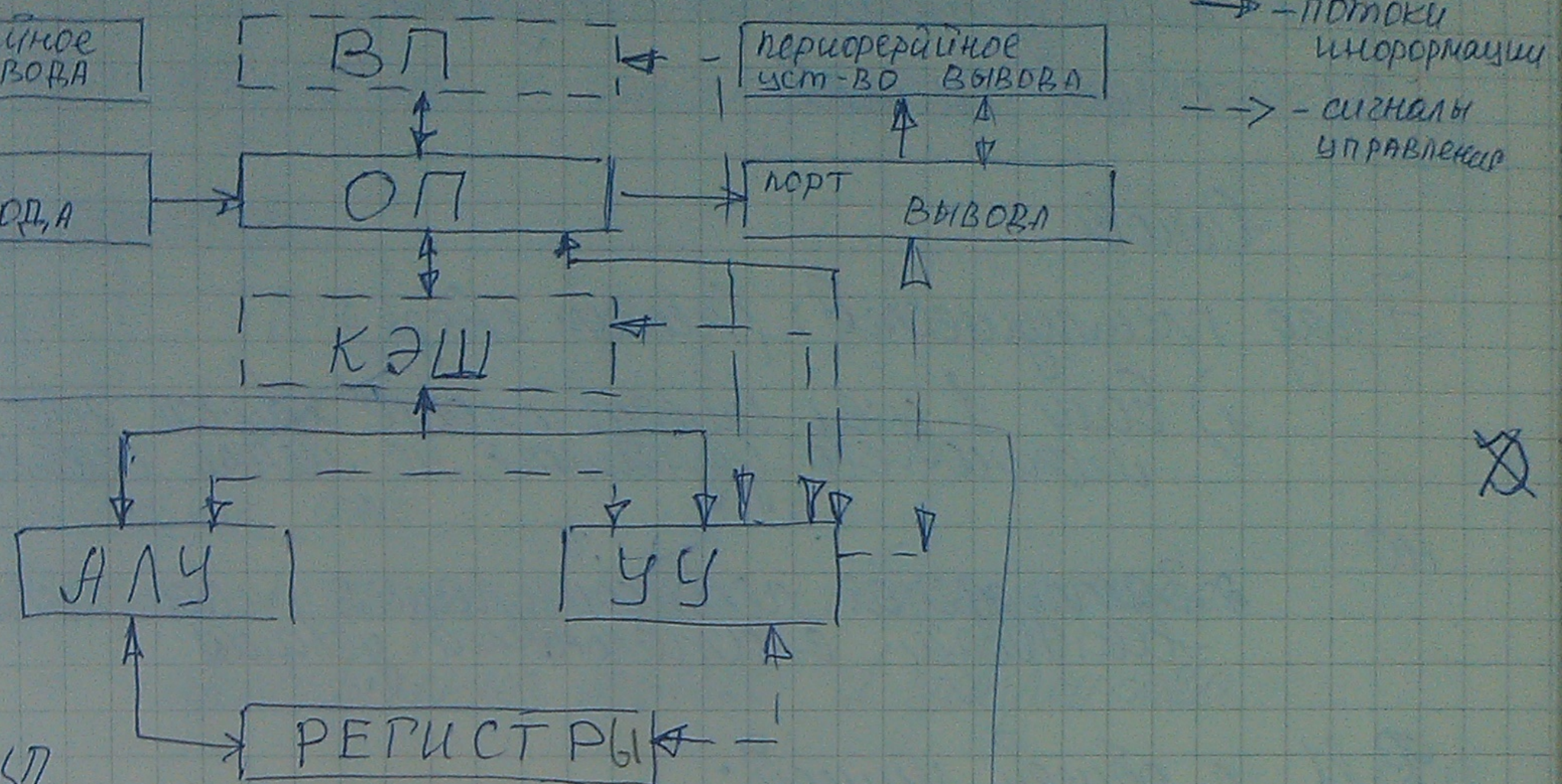
3) В концепции данные и команда в единую область памяти. Но

Общая память использ. в 1-ой машине в Принстонской

- Принстонская арх ⇒ единая память (больше исп.)
- Гарвардская арх ⇒ разделяемая память

4) Каждая ячейка имеет свой адрес, произвольный доступ к ячейке

Фон-Неймановская архитектура



ОП - основная память (оперативная) ВМ - Base memory

ВП - вторичная память (внешняя) ЕМ - external memory

КЭШ - КЭШ-память - СМ - Cash Memory

АЛУ - Арифметико-логическое устройство

УУ - устройство управления (CU - contral Unit

Основная память - состоит из ОЗУ: динамические полупров. ЗУ; работают медленно, но дешевы

КЭШ-память примерно в 10 р быстрее

УУ - чтобы машина работала в авт. режиме нужные коды в нужное временной послед-ти

Регистры, чтобы хранить промеж. рез-т. вычислений

ЦП - центральный процессор (CPU - central processor unit)

Структуры ВМ:

1). С непосредственными связями
2) С общей шиной

- Основа

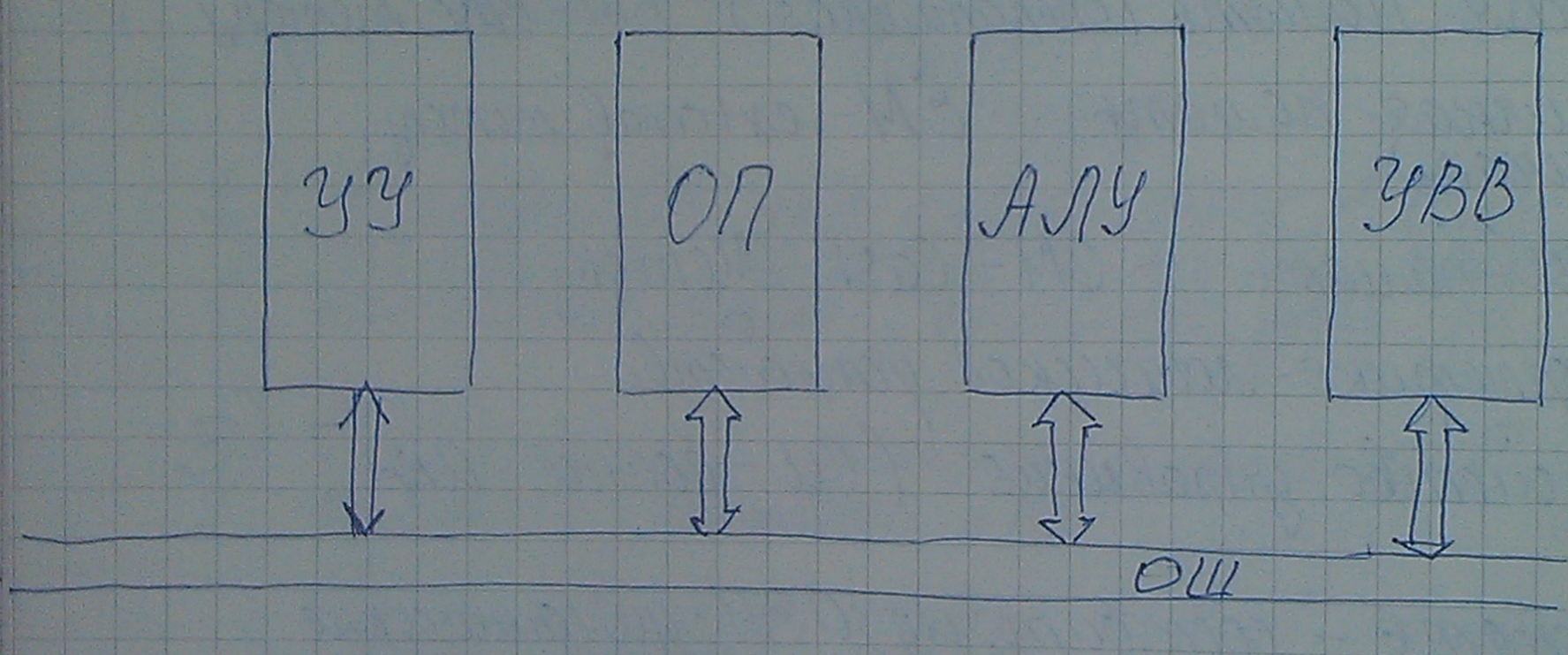
Т пад преимуществ: (непоср связь)

1) Если 1 узел имеет плохие хар-ки, это можно его улучшить, не меняя остальные

но

- Затруднена реконфигурация, надо менять все связи с остальными узлами

ВМ с общей шиной:

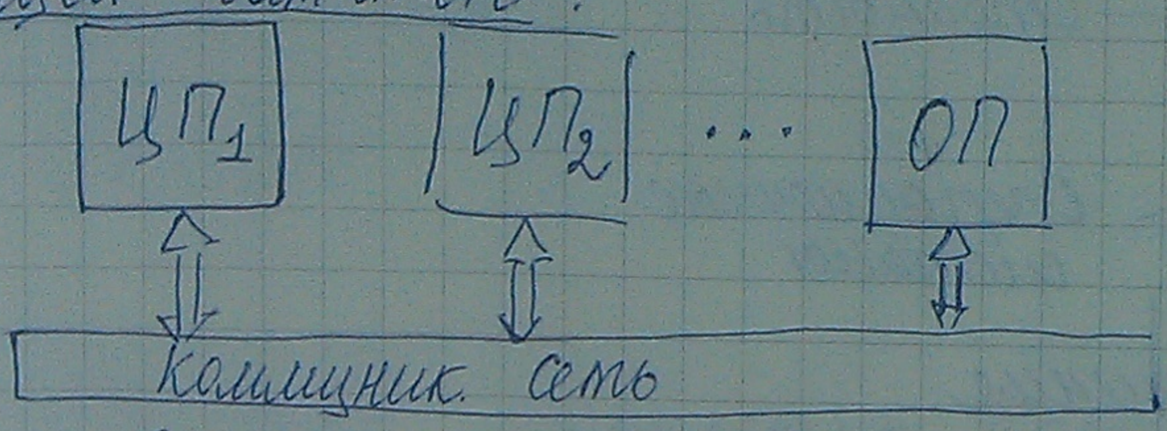


+ удобство реконфигурации
- информацией с ОШ может обмениваться в опр момент времени только 1 устройство

Структура вычислительной системы:

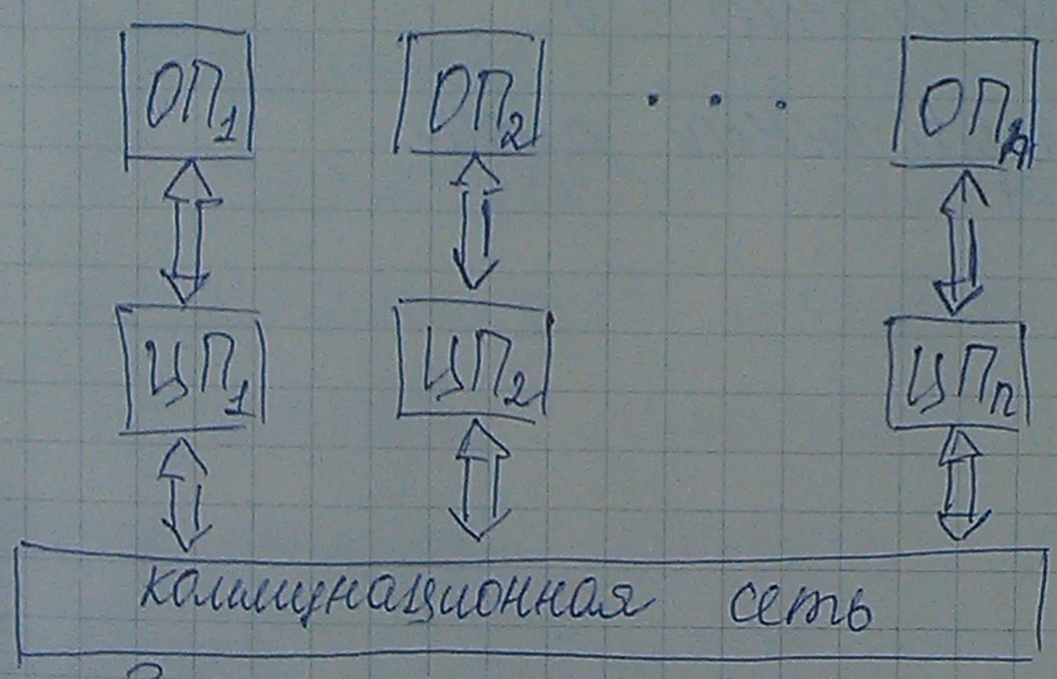
1) структура с общей памятью
2) структура с распределенной памятью

С общей памятью:



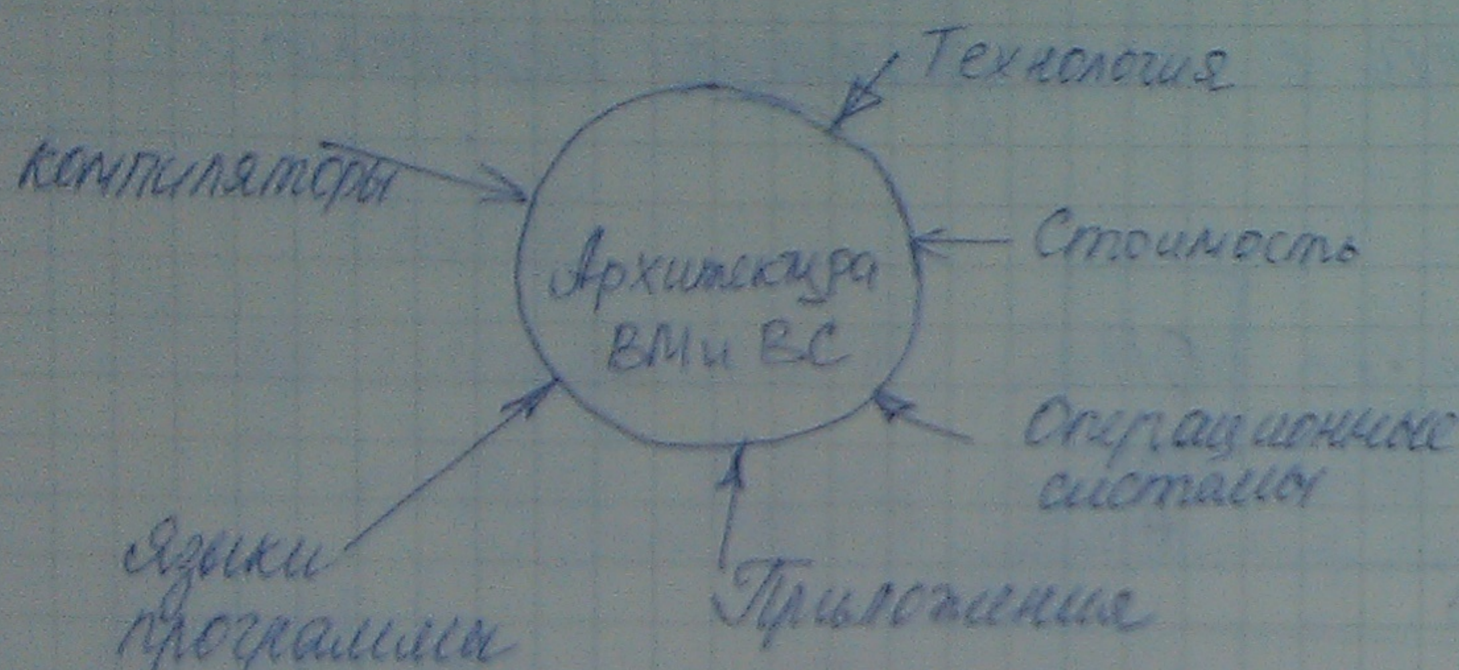
Часто вместо коммутир. сети – общая шина
– одновременно только 1 устройство

С распределенной памятью:



Каждый ЦП имеет собственную ОП
ЛВС – локальная вычислительная сеть
ГВС – глобальная вычислительная сеть
Машины обмениваются спец. сообщениями (пакетами)
ATM – самая быстродействующая из глоб.
Frame relay.

Перспективы совершенствования Вычислительных машин и систем (ВМ и ВС)



Технология - Больше кол во транзисторов на кристалл

Иерархия памяти

Компилятор должен выполнять команды, которые наиболее быстро выполняются одновременно

Эволюционные и революционные пути развития архитектуры

Классификация по составу и сложности команд: | Лекция 2

1) Архитектура с полным набором команд (CISC - Complex Instruction Set Computer)
2) Архитектура с сокращенным набором команд RISC (Reduced Instruction Set Computer)
3) Архитектура с (VLIW

В CISC-арх. семантич. разрыв пределов. засет регистр. системы команд, обще систем. команд, семант. анализиров. ЯВУ

Параметры АСК:

1) наличие большого числа команд, большая часть реализ.
2) наличие в прц. небольшого числа РОН
3) разнообразие способов адресац
4) множество команд с большой разр.
5) иногда обработка совмещ. с обращ. к памяти

можно отнести большинство ПК, г.в. 8086

Приводит к увеличению аппаратуры

Только 10% кода относится к сложным командам, но из-за них практически удваивают аппаратуру

Позднее время активно развивается RISC

1) ограничение числа команд только простейшими, наиб. часто встречаемыми

Эти команды должны работать в основн. регистрах проц., а обращение к основн. памяти - с помощью спец. команд.

2) уменьшается формат команд и способы адресации

3) аппаратн. средства проще
быстродействие выше

Реализация сложных команд с помощью простых быстрых RISC команд не менее эффективна, чем аппаратн. выполнен сложных команд в CISC.

Компании Intel и AMD стали использовать эл-ты RISC-арх. в своих проц-х

Различия между CISC и RISC стираются, появл. суперскалярные

Помимо CISC и RISC получила распр. архитектура с очень длинной командой
Концепция VLIW базируется на RISC
На качестве команды идеал из нескольких RISC-команд, они должн. вкл. в ком. слово.
Выигрыш в том, что многие команды выполн параллельно

Сравнительная таблица

Характеристика	CISC	RISC	VLIW
Длина команды	варьируется	единая	единая
Расположение полей в ком-де	варьируется	неизменное	неизменное
Доступ к памяти	Выполняется как часть команд разл. типов	спец. команды	спец. команды

Классификация по месту хранения операндов.

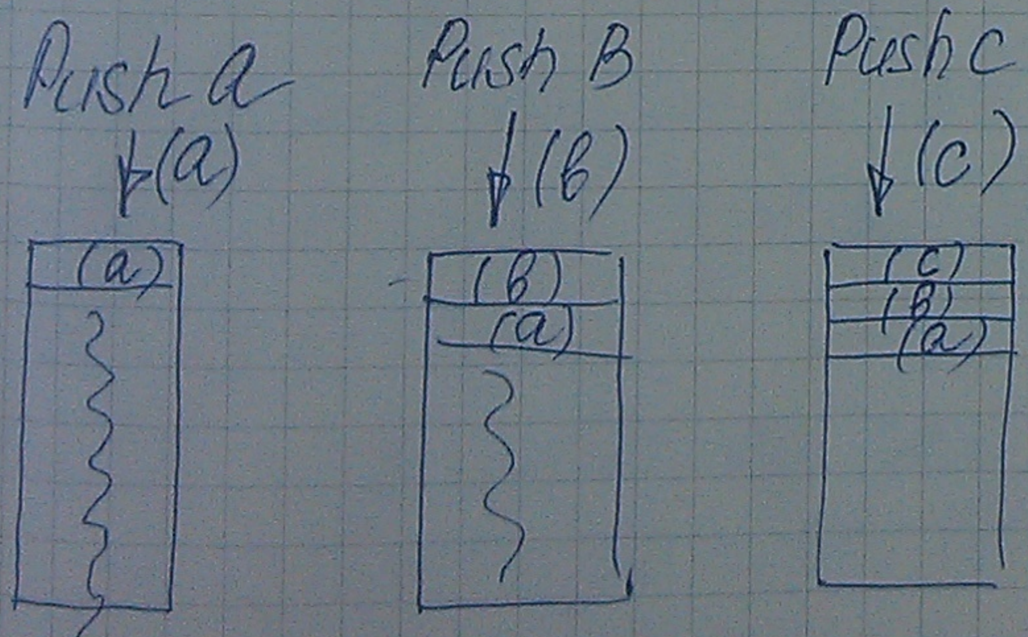
1) Стековая
2) Аккумуляторная
3) Регистровая

Выбор той или иной арх. влияет:

- сколько адресов б. содержать адресная часть команд
- насколько просто б. происходить доступ к операндам
- какой б. длина команд

Стековая

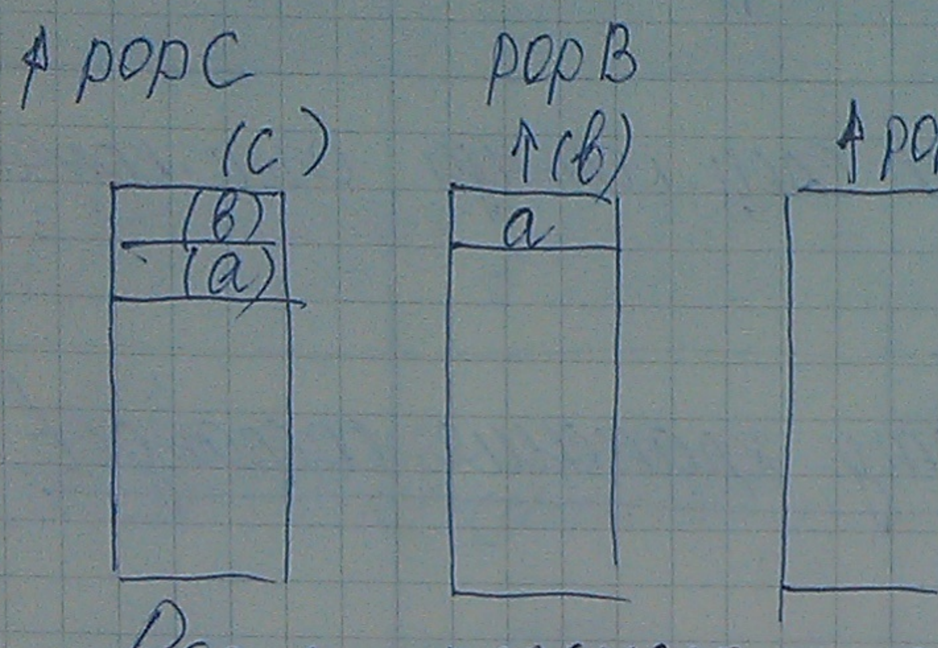
"множество логически взаимосв. по принципу LIFO



Push a - занести в стек команду с адресом a-

Если следующей будет push B, то в стеке произойдут следующие:

1) содержимое вниз
   (a) - на второй позиции

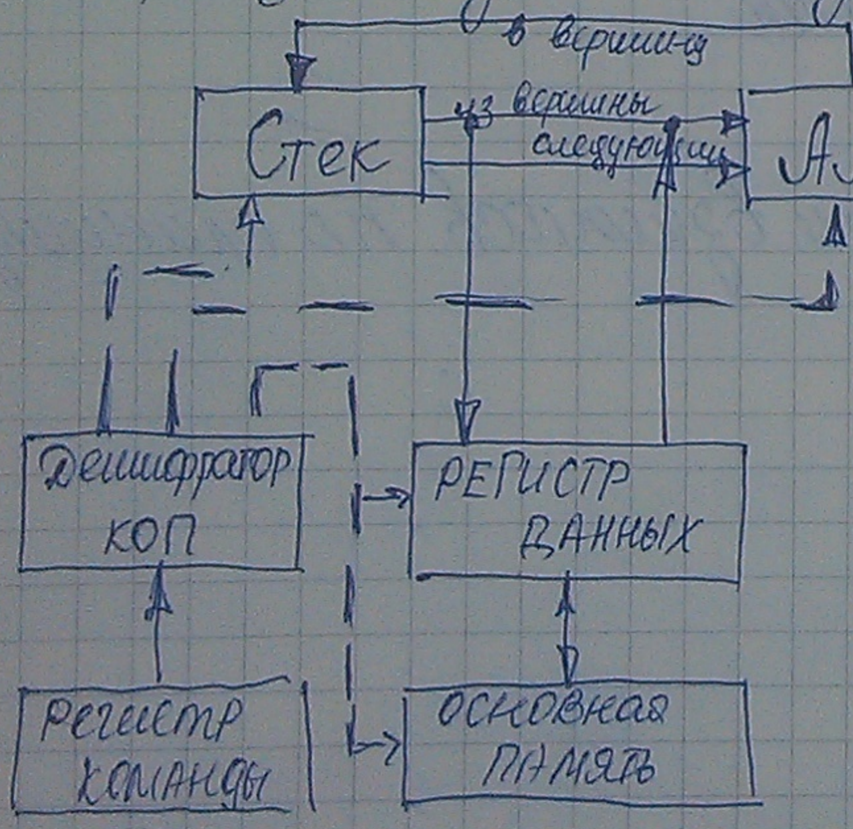


Pop c - перенести содержимое вершины стека в переменную с
на вершину стека (B).

Запись возможна только в верхн. позицию стека при этом содержимое сдвигается вниз на 1 ячейку

При чтении считанная инф. удаляется из стека, а содержимое верх. сдвигается

В машинах операнды перед обработкой помещаются в 2 верхние яч. стека после выполн. операции данное запис. обратно в стек



Информация заносится из АЛУ основной памяти передается в регистр данных и только затем в вершину стека.

Операции на 2 операндами

1) из вершины стека в АЛУ
2) из ячейки в АЛУ

Результат из АЛУ в вершину стека аналог. Сохранение вершины стека в ячейку с адр (х) – рор х

По этой команде содержимое верхней ячейки стека в ячейку (х), после чего все содержимое стека проталкивается на 1 ячейку вверх

⊕ – возможность сокращения адресной части команд

Поскольку все операции через вершину стека, адреса операндов указывать не нужно.

Код команды компактный, упрощается декодирование

⊖ Стековая архитектура не позвол. обращаться к памяти произвольным образом

Очень трудно создать эффект. трансл. код.

⊖ плохое быстродействие

Долгое время стековая арх. считалась неэффект. но в последнее время к ней проявл. интерес, связ. с популярностью языка Java, к семантике которого близка стек-арх.

ROS – машина со стековой арх. (Reduced Operand Set Computer)

Безопрандовые команды

Аккумуляторная архитектура

исторически возникла одной из первых

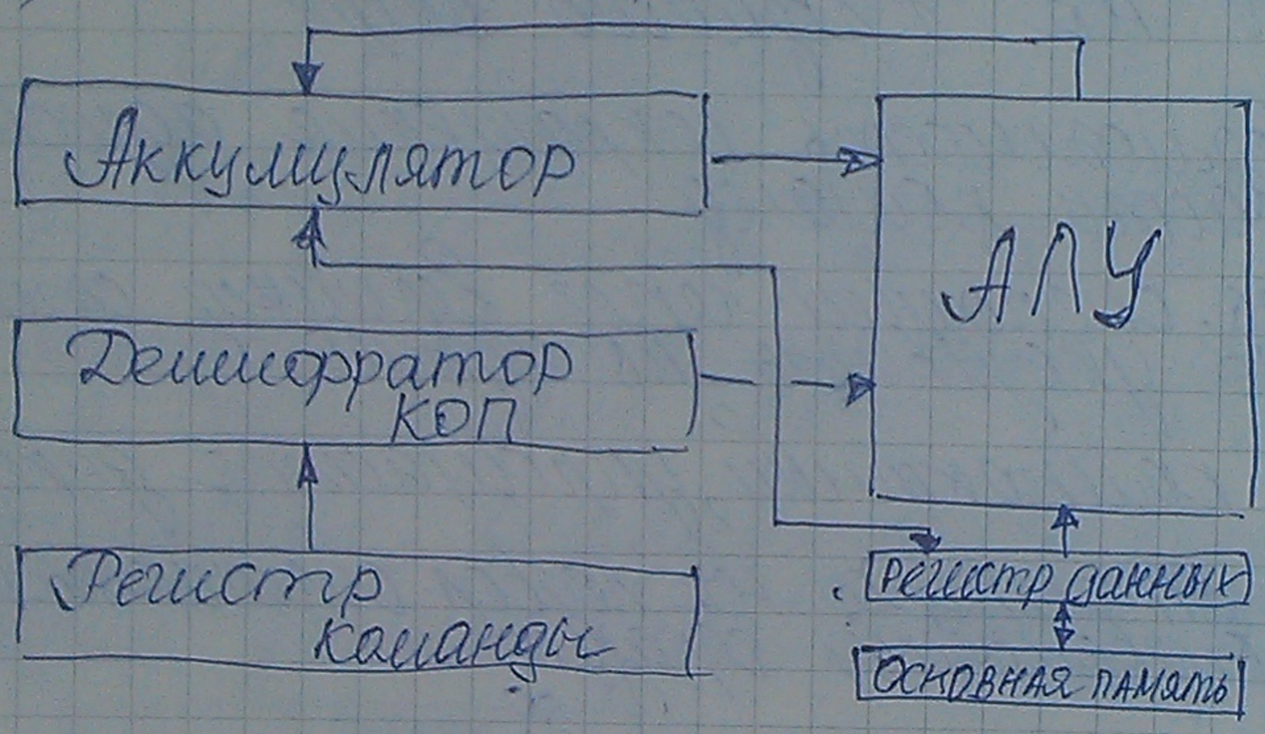
Для хранения 1 из операндов арифм. или лог. операции используется специальный регистр в процессоре, назыв. аккумулятором.

В этот же регистр и рез-т операции.

Поскольку адрес 1 из операндов предопределен, в ком. достаточно 1 из операндов

Изначально оба в основной памяти, 1 нужно загрузить в аккумулятор.

Если рез-т не является операндом для след. команды, его треб. сохранить в основной памяти



При загрузке в аккумулятор ячейки из ОП с адресом (x)
инф. счит. в ячейку с адр (x)

Вых

Запись в ячейку → Store X – выходом аккумулятора подкл. к шине информация из аккумул. в яч. (x)

Производится считыв. из памяти в RDB данных второй операнд находится в аккумуляторе
RDB рег. данных в АЛУ по окон. рез-т с выхода АЛУ записывается в аккумул.

⊕ короткие команды

⊕ простота декодирования

⊖ наличие только 1 рег → множество обращений к основной памяти

медленно

В настоящее время почти не используются

Регистровая архитектура

процессор включает в себя массив регистров общего назначения (РОН)
разрядность совпадает с разр. данных
к РОН можно обратиться, указав его номер.

- в процессорах с арх. CISC кол-во РОН небольшое:
  РОН → 8-32
  Адресная часть команды от 3 до 5 разр
  Можно указать 2-3 РОН в адр. части
- RISC арх. предусматр. большое число РОН (до нескольких сотен)
  Команды в RISC → 32 р. длина
  В команде не более 3 РОН

Регистровая арх. предусматривает хранение операндов не только в РОН, но и в памяти
Возможные операции:

Рег - Рег

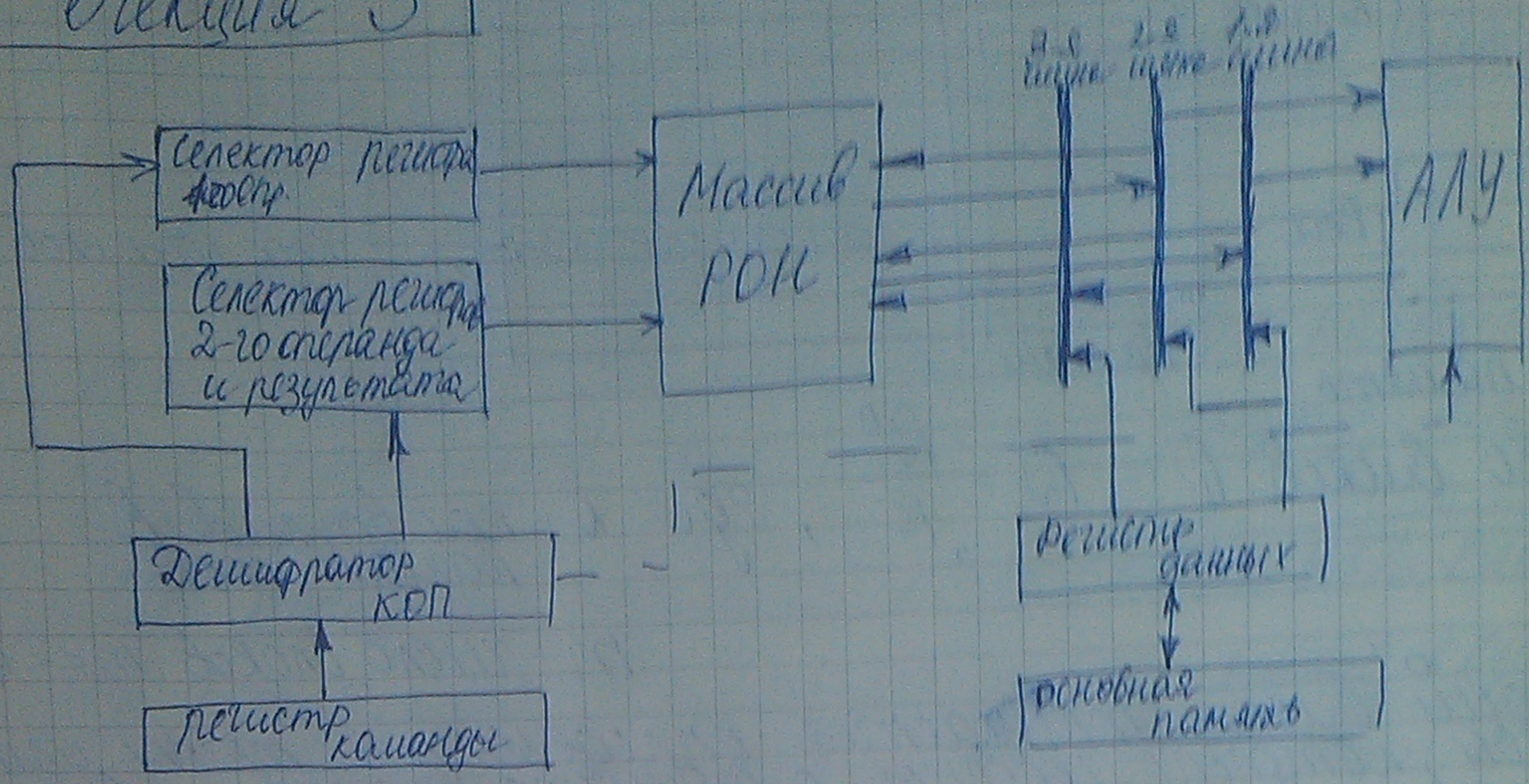
Рег - Память

Память - Память - редко, ибо медленно

Сравнительная таблица

Вариант	Достоинства	Недостатки
регистр – регистр	+ простота реализации + фиксир. длина команд + большинство команд за фиксир. кол-во тактов + простота компиляции	– большая длина обращения – часть разрядов не используется
регистр – память	+ данные не обязательно загружать в регистр + простота кодир. + объектный код компактный	– длинное поле адреса в памяти сокращ. место на адрес регистра ⇒ невозможность увязать много ро
Память – память	+ компактность пр. кода + не нужны регистры для хран. данных	– низкое быстродействие

Лекция 3



- Определение располож. 1-го операнда – рег. или памяти.
  Выбор рег. 1-го операнда или счит. 1-го операнда из памяти.
- Опред. располож. 2-го операнда – регистр или память
  Выбор рег 2-го операнда или счит 2-го операнда из пам.
- Подача на вход АЛУ операндов и выполн операций.
- Определение располож. результата – регистр или память
  Выбор регистра результата или ячейки памяти и запись результата операции из АЛУ.

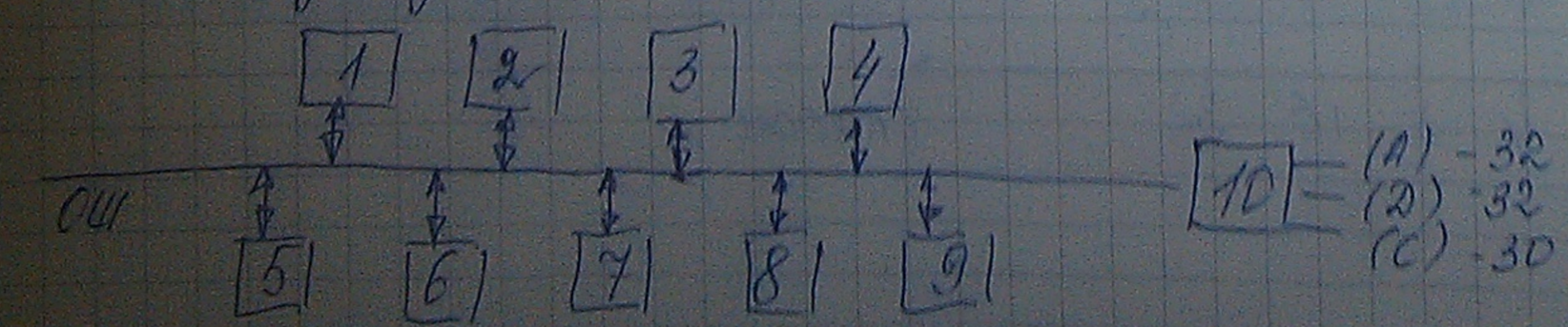
  3я шина служит для передачи из АЛУ в РОН или память.

Достоинства: неслож. кода, высокая скорость выполнения

В наши дни этот вид архитектуры системы команд считается преобладающим.

Микропроцессор Intel i486

Структурная схема



1) Блок АЛУ - произв. операции с фиксир. точкой, сдвиги лог операции

2) Блок УУ(СУ) - генератор тактовых импульсов, дешифр. упр. сигналов счетчик команд (IP), регистр флагов, дешифратор команд,

3) РОН или регистровая память (RGM)

4) Блок обработки данных / сопроцессор для обработки данных с плав. точкой

5) Буфер команд

6) КЭШ - память

7) Блок сегментации - сегментные регистры и регистры-дескрипторы

8) Блок управления страницами ; TLB - буфер трансляц.

9) Блок защиты памяти

10) Интерфейсный блок - связь процессора с общей шиной

Выполнение команд в процессоре i486

Процессор последовательно выполняет команду за командой. Номера представ. ся команд. работа прцсс по такам

Такты формир. генератором тактовым имп. в УУ

(A) + (M) → (M)
по адр II операнда

Послед. действий следующая

Такт 1 (выборка команд из памяти)

Передача адреса из списка команд в память (выдается на адрес действия а виртуальный адрес, т.е. ∃ спец. устройство преобр. вирт адреса в физические)

IA → EA
↑
instruction address — Executed address

Выбран RGA + содерж. ячейки памяти поступят в регистр команд УУ

eM → RGI

Такт 2 Декодирование кода и определение адреса второй команды в память

Определ. код операции

IA → EA, пересылает чтение содерж. ячейки памяти

Такт 3 Выборка операнда из памяти

Read (M) → RG2 АЛУ

Другой операнд пересыл. из аккумулятора в RG1 АЛУ

(A) → RG1 АЛУ

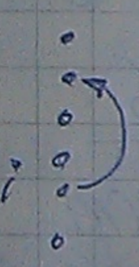
Такт 4 Выполнение операции сложения в АЛУ

Результат в аккумуляторе

Такт 5 Результат операции, произв в АЛУ → M

Рассмотрим пример выполн. команд безусл / усл. перехода

УП



Такт 1 Выборка команды

Такт 2 Расшифровка команды

Такт 3 Выборка операнда

Такт 4 Выполнение операции в АЛУ

Анализ состояния флага в зависимости от результата выполнения операции

Если f = 1, то в СТ записыв. адрес условного перехода;

Если f = 0, то след. командой будет i + 1

Видов исп. циклов попадка 10

Замечание 1. В первых тактах действия одинаковы для всех команд

Формир. кодов содерж. СТ и Б сделано достаточно поздно, далее в такте T2.

Замечание 2 выполнение команд в каждом такте осущ. управл. шинами, кот. вырабат. на пред. дешифр. кода операции (T3, T4, T5)

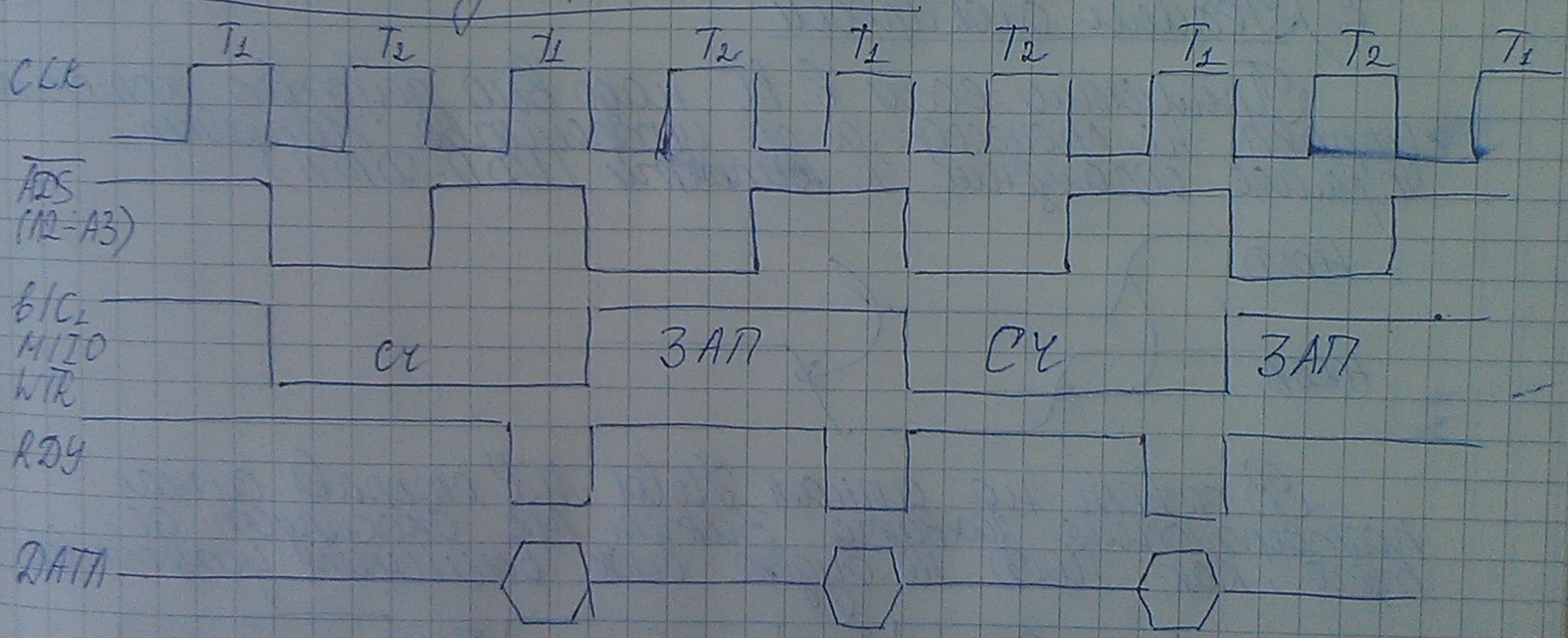
Замечание 3 можно начать выполн. след. команду, не дожидаясь завершения предыдущей

Конвейер. использ. для ускорения вып. команд засчет одновр. выполнения неск. операций

В данном примере выгодно - конвейер на 4 такта, т.к. не всегда результат нужно помещать в память

Замечание 4 При выполн. некоторых команд

Рассмотрим вр. диаграммы i486



CLK - тактовый импульс

$\overline{ADS}$ - Address Status - адрес действителен

D/IO либо данные, либо устр. вв/выв

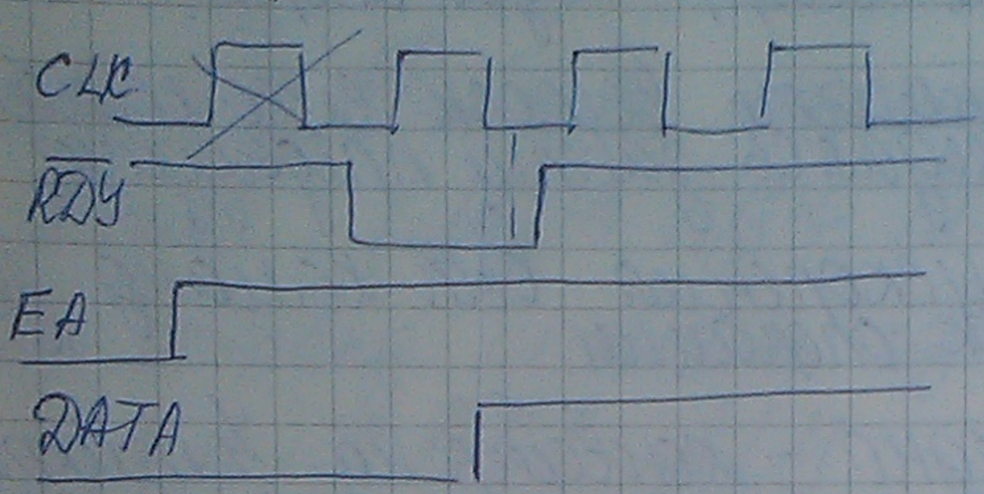
W/R - запись / чтение

D/C - данные / упр. сигналы

$\overline{RDY}$ - сигнал готовности

Особые режимы в работе i486

1) Режим ожидания. Необх. иногда возникать, когда идет обращение к мед. устройством
Необходимо, чтобы АЦП уст. низкий уровень сигнала READY

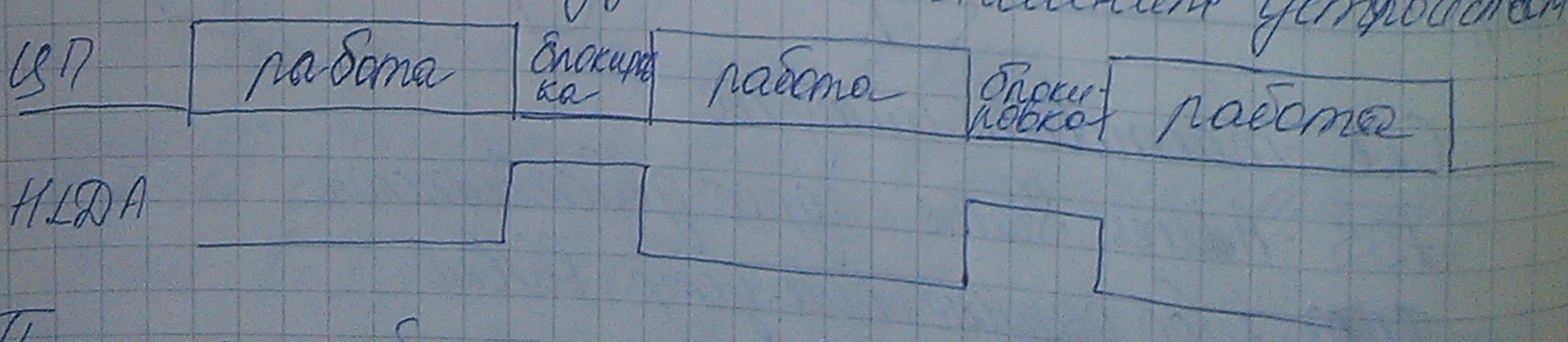


УП выставил EA и готов принять данные, то АЦП не готово передать данные и RDY сигнал высокий уровень
Данные появ. только после того, как READY обратится во

2) Режим блокировки

Применим, если в ВС надо вып. действия, которым управл. не процессор, а др. устройство. Напрямую прямое обращение к памяти (ПДП-DMA)

В ответ на сигнал HOLD МК отправ. сигнал подтверждения захвата. Затем МК блокирует до того, как HOLD не будет снят внешним устройством



Пусто для обращ. нужно 2 такта. В этих случаях процессор будет блокирован на эти 2 такта

Лекция №3 - переписать

Лекция №4 Представ

Кол-во разрядов	Название
1	бит
8	байт
16	слово
32	двойное слово
64	учетверенное слово
128-512	строка, слог
1024-16к	страница

Числа с фиксированной точкой

Дробные числа масштабир. и переводятся в целые
Необходимо следить, чтобы не произвн. разр. сетка

Числа с плавающей точкой.

32-разрядные - с одинарной точностью
64-разрядные - двойная точность
80-разрядные - расшир. точн.

Числа корректир. при вводе и выводе. (увелич или уменьш.

В проц. фирмы Intel все операции с плав. точ. выполн в сопроцессоре

В нем все действия с 80р. операндами

Если треб. одинарная или двойная точность, при выводе числа округляются,

Представление команд в ВМ

Код	А

Команда это код, который выражает
или в той или иной степени

Классификация команд.

- по функцион. назначению
  - а) команды пересылки данных
    - регистр ⇄ регистр
    - регистр ⇄ ВМ
    - регистр ⇄ УВВ.
  - б) команды преобразования данных
    - арифм.
    - логические
    - сдвига
  - БП, УП – команды установл. прог. в опр. режим
    - установка и сброс флага

Структура команды

Формат команды – структура команды

ЯВУ ⇒ Компилятор ⇒ Языки Ассемблера ⇒
⇒ Команды ВМ ⇒ Микрокоманды

- CISC - полный набор - от 100 до 1000 команд
- RISC - сокращ. набор - от 30 до 100 команд.

Способы адресации:

IP - instruction pointer (16 разрядный)

EIP - 32 разрядный → 4 ГБ памяти
extended

IA - instruction address - адрес, указ. в команде

EA - исполнительный адрес (физич.
Executed

Преобр. адреса из программного в испол. адрес.

1) Непосредственная адресация

операнд указывается непоср. в команде.

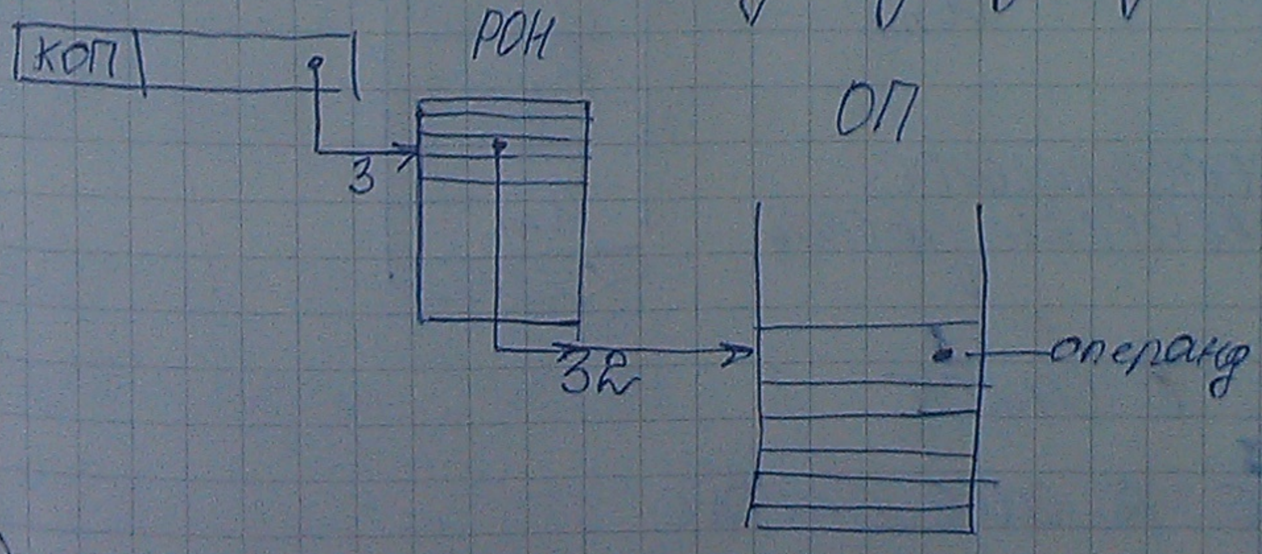
2) Прямая адресация

в команде указ. адрес операнда
адрес РОН 32 разр. - 8 лег.
4 разр. - 16 лег.

адрес операнда в ОП - невыгодно, слишком длинный путь команд

3) Косвенная адресация

в команде указ. адрес адреса операнда



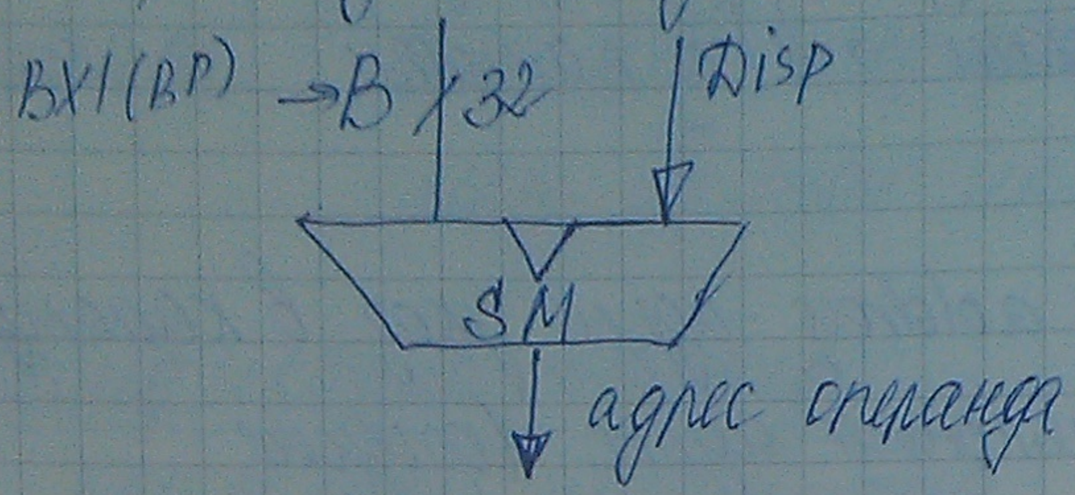
4) Базовая адресация

данные располож. в разных местах
часть адреса, кот. указ-ся в команде
назов Dysp. - смещение

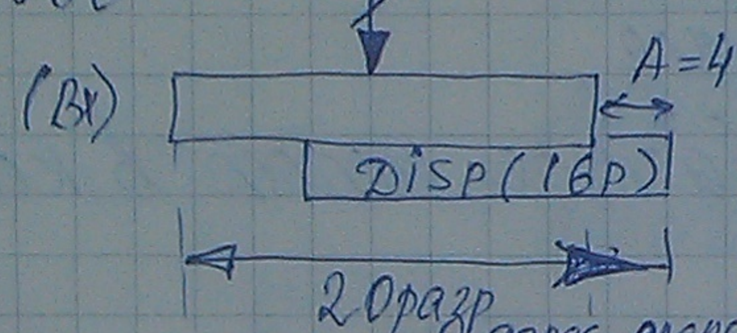
3 способа ф-ции исполнительного адреса ЕА

а) базовый адрес и смещение соединяются (конкатенация)

б) базовый адрес и смещение суммируются



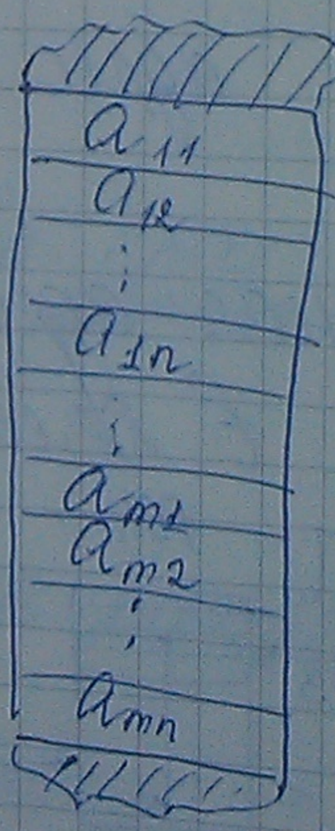
в) промежуточный случай в 8086



Сдвиг (BX) на 4 разр. влево

5) Индексная адресация

$$\begin{matrix} a_{11} & a_{12} & & a_{1n} \\ a_{21} & a_{22} & & a_{2n} \\ - & - & - & - \\ a_{m1} & a_{m2} & & a_{mn} \end{matrix}$$



Для обработки содержимое индекса I автоматически увелич. на

(I)+1 – переход в строке

(I)+n – переход к след. строке

SI, DI

6) Общий случай адресации

$$EA = (B) + (I) + Disp$$

32 разр. 32 разр. 32 разр.

Частные случаи:

базовая : $(B) + Disp$

индексная : $(I) + Disp$

базово-индексная : $(B) + (I)$

базово-индексная со смещ. : $(B) + (I) + Disp$

Индексная адресация с масштабированием

$$I = (I) \cdot F$$

F - масштабирующий коэф-т.

$F = 1, 2, 4, 8$

Знач. F →

$$SS = \begin{cases} 00 - \text{сдвиг нет}, & F=1 \\ 01 - \text{сдвиг на 1р}, & F=2 \\ 10 \quad\quad\quad\quad 2р., & F=4 \\ 11 \quad\quad\quad\quad 3р & F=8 \end{cases}$$

Формат команды i486

Основная часть команды без префикса.

Название полей	КОП	MOD R/M	SIB	Disp	Data	Сам операнд в случае непосредственной
Число байт	1,2	0,1	0,1	0,1,2,4	0,1,2,4	

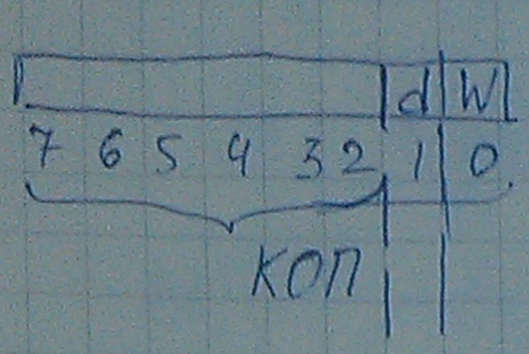
КОП - код операции

MOD - Модификатор

SIB - Scale Index Base, масштаб инд. базы

КОП

команды м.б одно или 2 адресные



W - разряд длины данных

W=0 → 8 разр. данные

W=1 → 16 или 32 разр данные

D - дескриптор

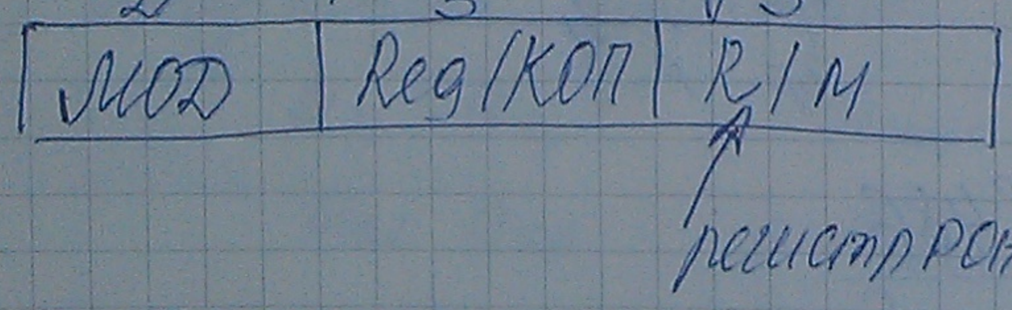
D=0 - 16разр. данные

{} D=1 - 32разр. данные

MOD R/M

спецификатор регистр/память

-байт режим адрес



AX - аккумулятор

CX - счетчик

BX - базовый индекс

SP - указатель стека

BP - базовый адрес

SI - индекси регистро

DI - " —

DX - счетчик

1102 = { 00 - смещ.-е СК
01 - длина поля смещ. 1 байт
10 - 2 байта или 4 байта
11 - вып-ая команда (R)*(R)

Если команда 2-адресная, 1 разряд КОП d → рез-т операции

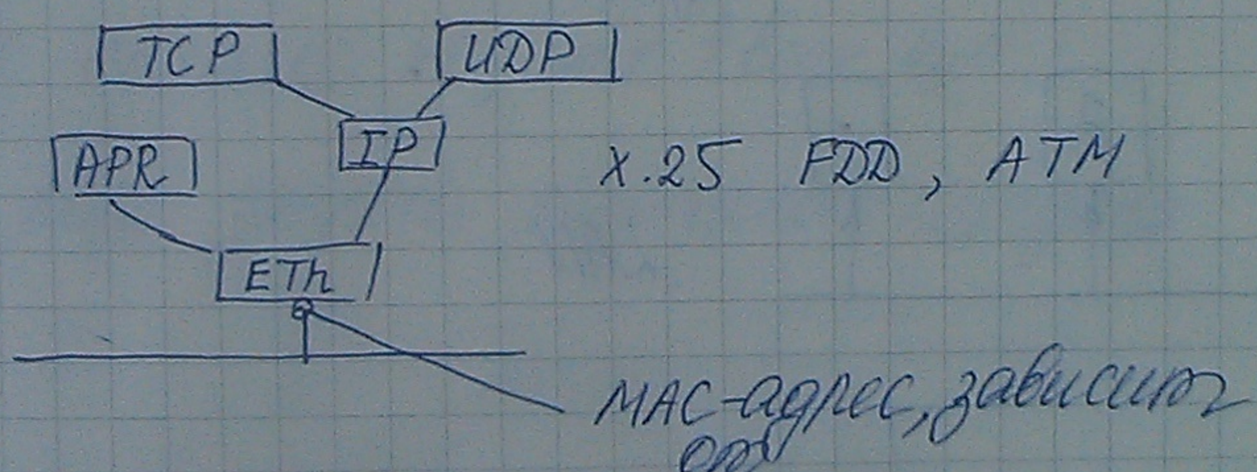
d=0 ⇒ результат помещен в RM, т.е по адресу 2-го операнда

d=1 ⇒ — // — — // — в Reg/КОП. по адресу 1-го операнда

Лекция №4 Сети О—О

Протоколы канального уровня.

1) Протоколы LLC
2) ARP, RARP

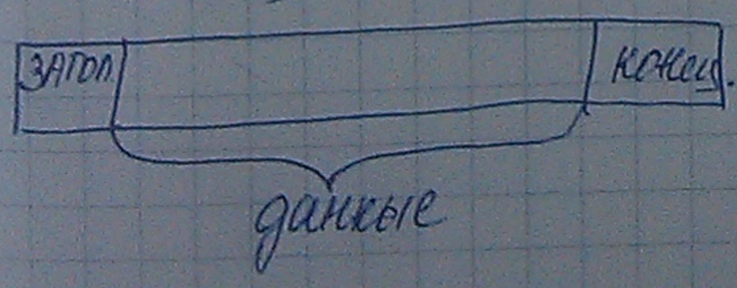


Проблемы:

- MAC-адрес - жестко зафиксир. в сетев.
IP-адрес - гибкий

На канальном уровне:

сообщение имеет вид кадр



Лекция №5 | ОЭВМ

SIB - Scale Index Base

2	3	3
SS	I	B

SS - масштабный коэффициент
F = 1, 2, 4, 8

I - адрес индексного регистра (SI или DI)

B - адрес базового регистра (BX или dx

Выравнивание команд:

- команды записываются в очереди без промежутков
- нужна спец.аппаратура, чтобы найти начало и конец команды
- в КОП записана информация о длине команды.
- (прочитав КОП, можно определить длину)

Как часто тот или иной способ адресации встречается:

1) Непосредственная адресация: 20% - 40%
2) Прямая адресация: 5 ÷ 20%
3) Косвенная адресация 1 ÷ 5%
4) Базовая адр. со смещением 35 ÷ 55%
5) Базово-индексная со смещением 0 ÷ 15%

Адресация данных разной разрядности:

Данные имеют длину кратную байту, хранятся с разрешением по адр., какой объем имеется ввиду.

МП Intel	ad	D
8080	16	8
8086	20	16
80286	24	32
80386, 80486 и далее	32	32

Адрес формируется A31-A1

16 разр. данные - A0 игнорируется

8 разр данные

A31-A1 - выбор 16 разр слова

МП и ст. байт выбирать A0 и BHE

16-разр. процессор

A0	BHE	
0	1	D8 - работать с 8-разр данными
1	0	
0	0	- D16

$A_0 = \overline{BHE}$

32-разр. процессор

$\overline{BE0}$	$\overline{BE1}$	$\overline{BE2}$	$\overline{BE3}$	
0	1	1	1	d8 - работаем с 8-разр. данными
1	0	1	1	
1	1	0	1	
1	1	1	0	
0	0	1	1	d16 - работаем
1	1	0	0	

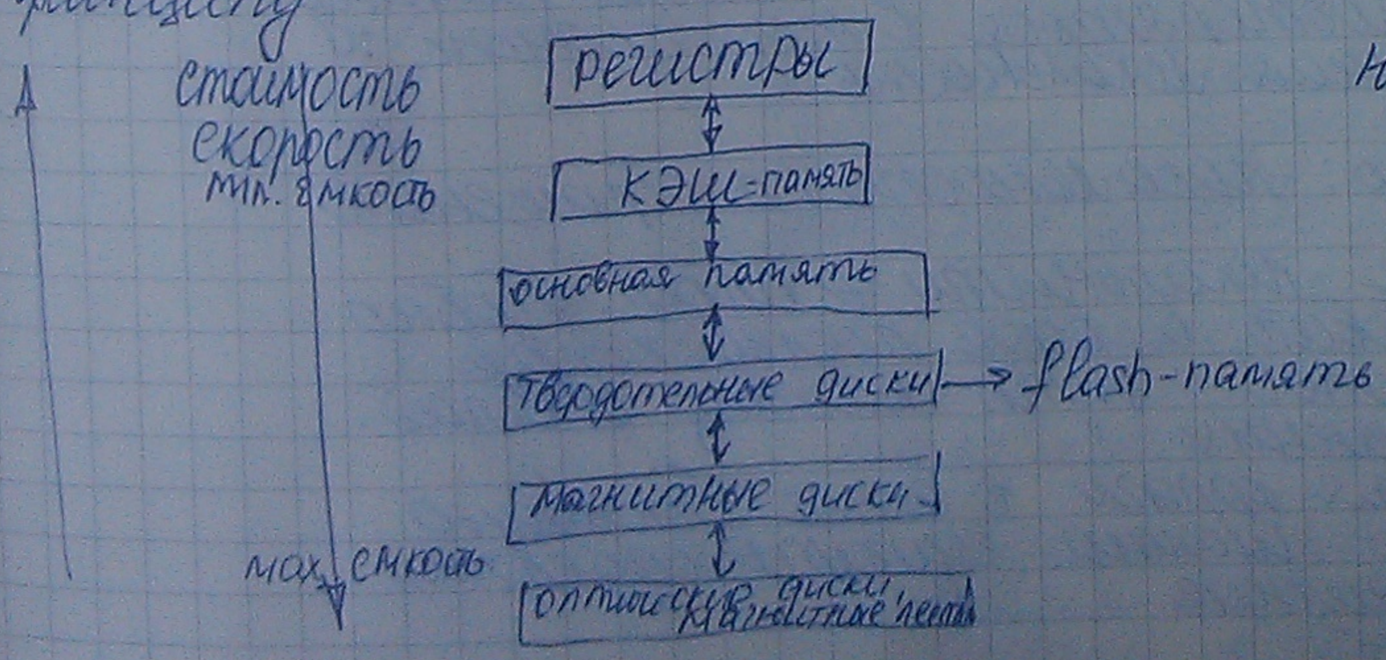
A31-A2 - адресная шина
A1-A0 - формируют $\overline{BE0} \div \overline{BE3}$

64-разр. процессор
адрес шина A31 ÷ A3
A2-A0 - используется для формирования сигналов $\overline{BE7} \div \overline{BE0}$

Иерархия памяти

Цель: обеспечение требуемой емкости и быстродействия за приемлимую цену

Наиболее распростр. подходы для решения этой задачи явл. построение памяти по иерархическому принципу



на каждом ур:

Информация в основном уровней полностью содержится в них

Внутр. память
- Регистры – РОН, самый быстрый сверхоперативное ЗУ наим. емкость.
- Кэш-память – на статич. ЗУ,
- Основн. память – на динам. ОЗУ и ПЗУ

Внешн. память
- Твердотел. – flash } вторичная память
- Магн. диски – винчестер }
- Опт. диски и магн. ленты – сменные носители емкость не ограничена } третичная память

Локальность по обращению

- с высокой вероятностью адрес очередной команды программы либо следует непоср. за командой, либо расположен вблизи него
- такое расположение адресов называется пространственной локальностью программы
- аналогично с данными: они расположенного массивами в смежн. ячейках: пространственная локальность данных
- как правило, в множество циклов и подпрограмм: небольшие наборы команд будут повторяться в теч. прогон. : временная локальность
- все это: локальность по обращению
- из св-ва локальности ⇒ прогр. удобно представить в виде послед. обраб. фрагментов: группа команд и обраб. данных. Помещая фрагм. в более быструю память можно снизить издержки на обращение

Команды и данные, будучи перед ЦП в быстрое ОЗУ, ср. вр. доступа опр-ся более быстрым ЗУ. Это позволяет хранить массивы данных на магн. дисках, а в процессе обработки использ. небольшую, быструю память

Информация на каждом уровне разбивается на блоки, пересылаемые между 2 соседними уровнями; объем блока обычно фиксирован и увеличивается при переходе от нижних ур. к верхним

Ex. B: основная память ⇔ магн. диски
кэш-память ⇔ основная
} обмен ≈10..Кб блоками ≈100 Б

При поиске информации она отыскивается сначала в верхнем уровне и обращение происходит к нижним только тогда, когда на верх. ур. информации нет

Факт обнаружения нужной информации — „попадание", hit.

В противном случае - „промах", MISS.

При промахе → поиск в более низком ур-не, где возможны hit и miss.

При промахе блок переписывается с нижнего ур. на верхний.

При оценке эффективности исп. характеристики:

1) Коэффициент попадания (hit rate) - отнош. числа обращений, при которых произошло „попадание", к общему числу обращений к данному уровню иерархии

2) Коэффициент промаха (miss rate) — " — " — " — при которых имел место промах, — " —

3) Время обращения при попадании (hit time) - время необходимое для поиска инф-и в памяти верхнего ур-я (вкл. доп. вр. на обращение попадания + время на факт. счит. данных)

4) Потери на промах (miss penalty)

- время, требуемое для замены блока в кашмяти более вых. ур. на блок с нужными данными, располож. в памяти более низкого ур
(вкл. время доступа - время обращ. к первому блоку при промахе + вр. передачи оставшихся слов блока

Основная память

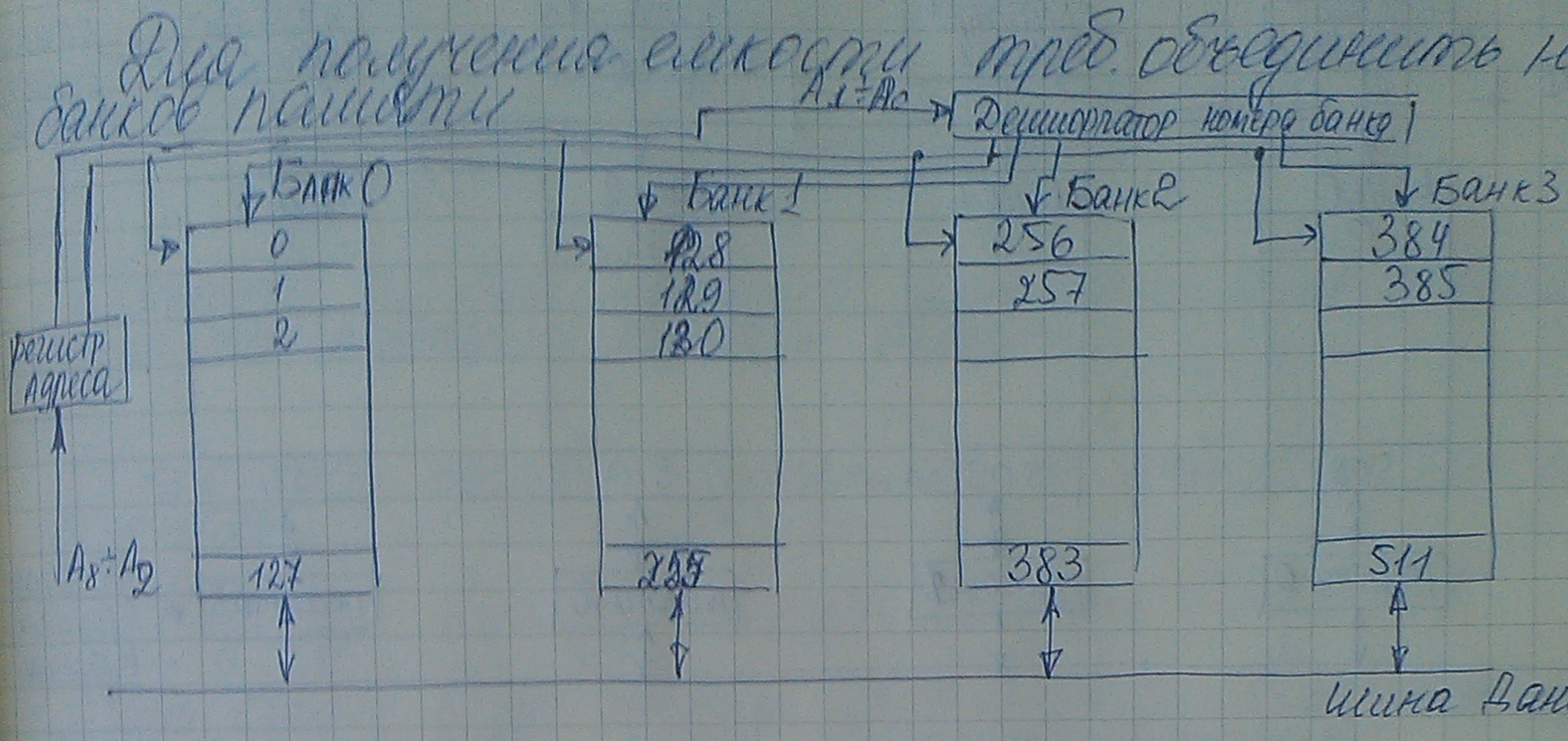
- единственный вид памяти, к которой СР может обращ. непосредственно (исключение регистры)
- информация, хран во внешних ЗУ, будет доступна проц., только после передачи
- образуют устройства с произв. доступом: обращение к ячейке памяти занимает одинак время и может произв в любом порядке.
- включает 2 типа устройств
  - ОЗУ (оперативные), - RAM
  - ПЗУ (постоянные). -
- принцип. одно образуют ОЗУ доступ как запись, так и считав информации
- ОЗУ = RAM = Random Access Memory
- для ОЗУ характ. энергозависимость: при кратковр. откл питания - информ. теряется
используются только как врем. память
- ПЗУ - энергонезависимые Read Only Memory (ROM = обеспечивает считыв. информации; изменение либо вообще нельзя, либо знач. дольше и по-другому

Лекция №6

Блочная организация памяти

Емкость совпадения

Для получения емкости треб. объединить несколько банков памяти



Младшие байты выбирают банк, старшие - внутри банка

Быстродействие - как у 1 банка, емкость как у Σ банков

Расслоение памяти (address interleaving)

Наличие св-ва накапливания блочная организ. обладает еще и „+", сокращ. время доступа

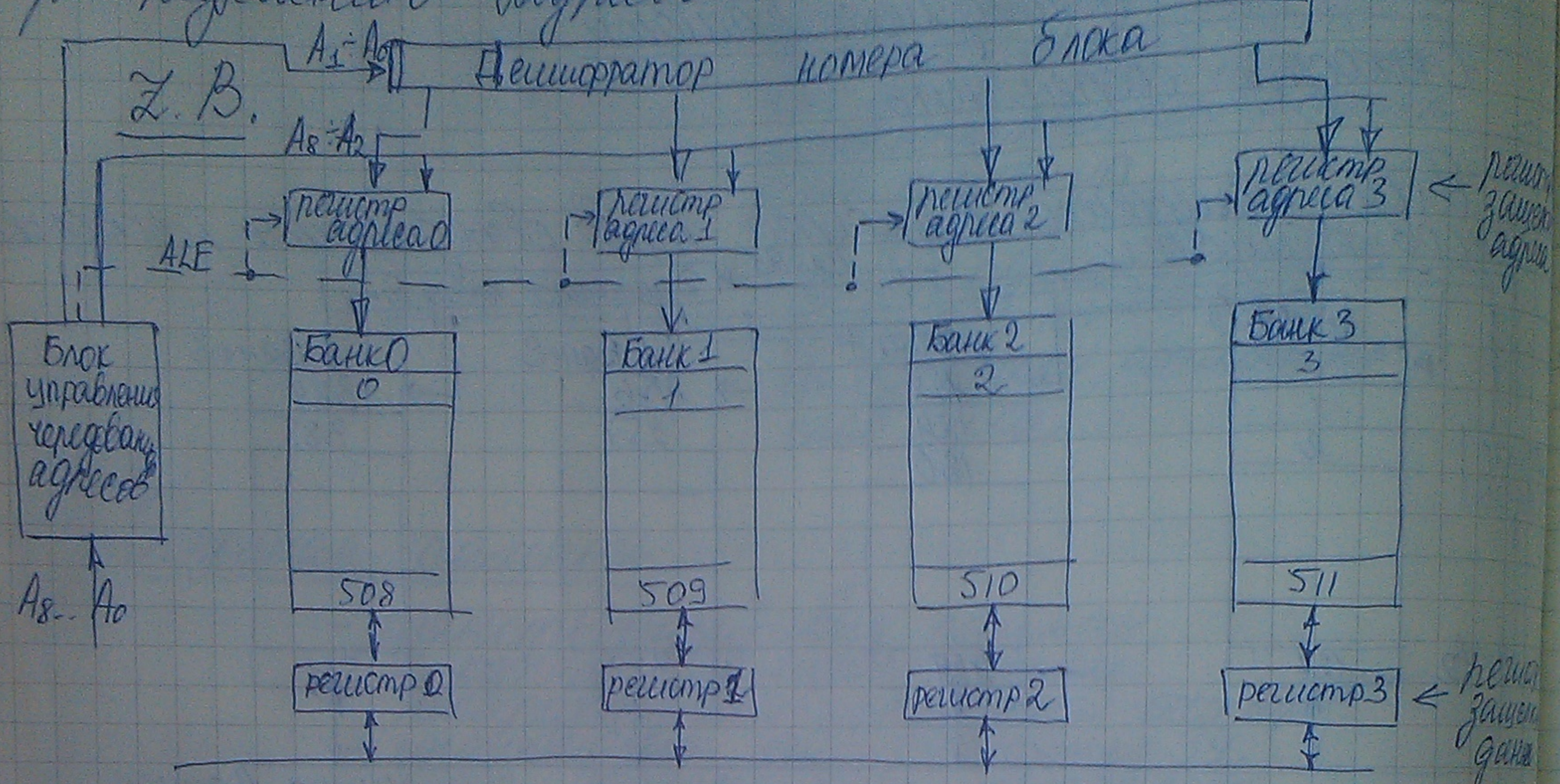
Это возможно благодаря блочной параллельности; одновр. доступ ко всем банкам

Одна из методик, называется расслоением памяти

– в ее основе лежит чередование адресов: распределение

Принцип "локальности" при обращении:

Чередование адресов возможно благодаря цикли.
распределению адресов



ALE - Address ~~Latch~~ Latch Enable

~~Один и тот же адрес~~

В 1-м такте адрес защелк. с помощью сигнала ALE

2-м
3м
4-м
} адрес защелк в 2-м рег.
3-м рег.
4-м рег.

Обращение к ячейке происходит независимо от того, что хранится в ней банке.

Удается за 4 такта обратиться к 4-м банкам (раньше к 1-му)

При большом кол-ве банков уменьшить время обращ. в 3 раза

Это верно при идеальном чередовании адресов обращ. след. к ячейкам банков, поочередно

Если нет обращения к i и тому же банку,
↓
конфликт по доступу
↓
необходимо ждать спр. время, пока не завершится обращение к банку
↓
уменьш. быстродействие памяти

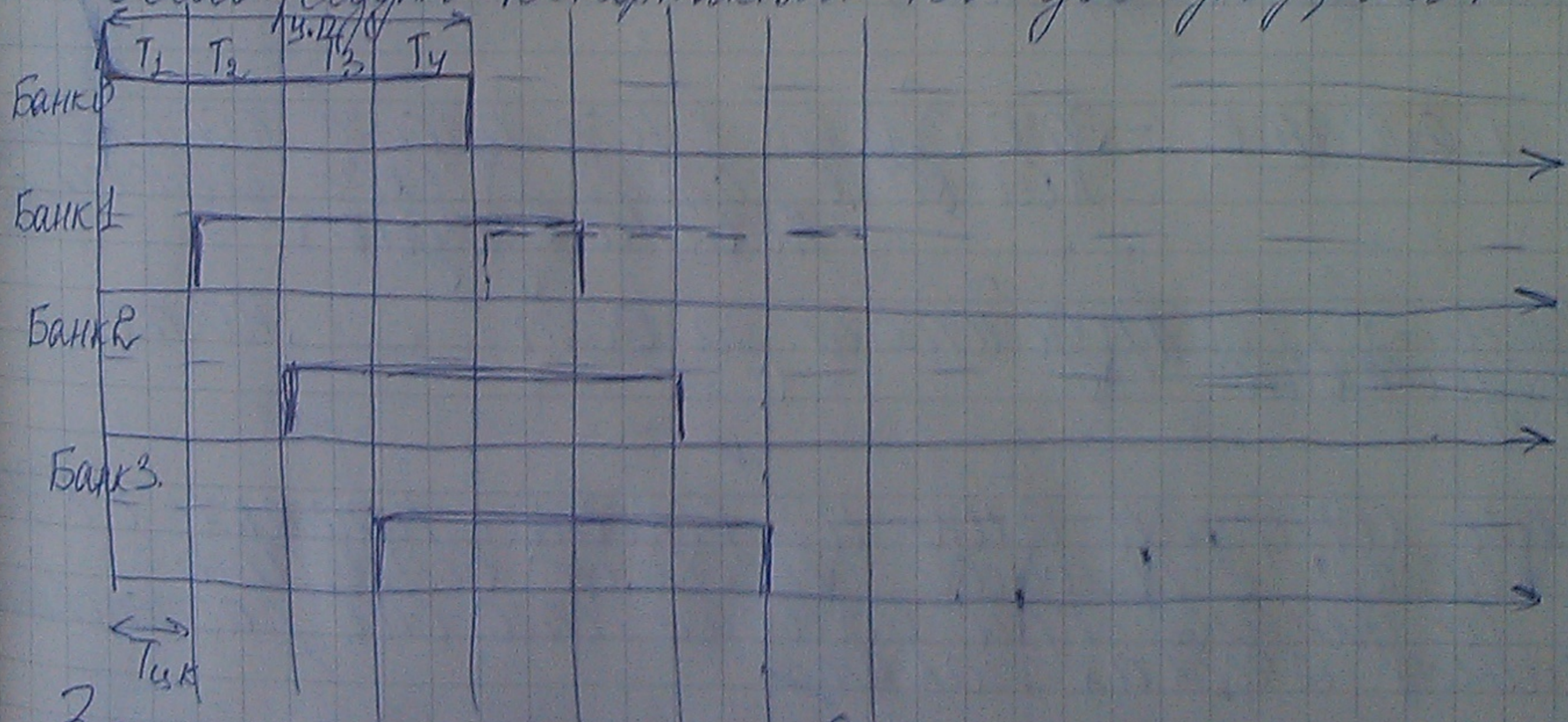
Лучшего из-за того можно сказать при большом B, что уменьшается вер-ть отказа

Блок упр. чередов. соединит ск. арбитра на осн. счетчика.

Каждому банку - 2 позр. счетчик - код 1, 1

При обращ. к др. банку вес (код). Сле. обращ. возможно тогда, когда счетчик обнулился

Если - сразу конфликты по доступу, wait сг = 0.



Запись информации в память

T_1: Сигнал подается в банк 0

Пунктиром - работа системы без расширения

$T_{ц.п}$ - время цикла памяти

$T_{ц.к}$ - Тцикла конвейера

Если B - кол-во банков

$$T_{ц.к} = \frac{T_{ц.п}}{B}$$

$t.S \Rightarrow B$

В дальнейшем при рассмотр. основной памяти будем считать, что она выполнена по принципу конвейерной адресации.

~~В сериях AMD Intel~~

Она поддерж. во всех сериях intel, начиная с i3.

Если происходит конфликт по доступу, то конвейер нарушается.

ЛБК последнего адресует к памяти ≈ 93%. Времени

В оставшиеся 7% случаев ищет место конфликта по доступу. Сейчас выпуск БИС памяти, где конф. решены внутри микросхемы. Они работают в 3÷4 раза быстрее.

КЭШ-память

КЭШ-память на статических ОЗУ (они работ. быстрее, но стоят дороже)

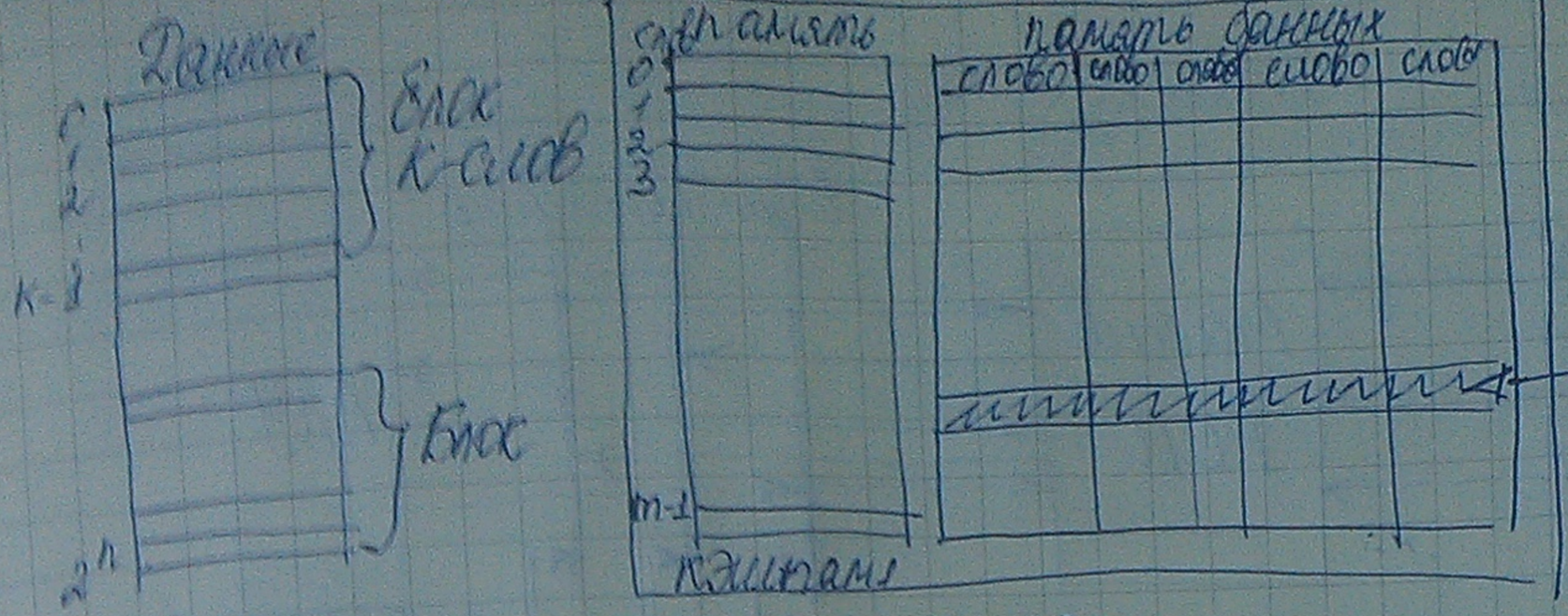
КЭШ-память – это буфер в нач. разряд. копир. бл. как-бы не разр. слово зависим. данных задач.

Выигрыш в быстродействии за счет ___ности по быстродействию

Если мы копируем блок из основной памяти в КЭШ, то мы можем с большой вероятностью выбирать команды и данные при условии, что они располож. в смежных ячейках памяти.

Программист может и не знать о суш. КЭШ-памяти. Когда СР пытается прочитать слово из памяти, то сначала осуществляется поиск этого слова в КЭШ-памяти. Если она существует, то слово извлеченное из КЭШ-памяти. Такую ситуацию принято называть попаданием. При промахе весь блок переписывается из основной памяти в КЭШ.

Рассмотрим структуры обращения к КЭШ-пам.



Осн память

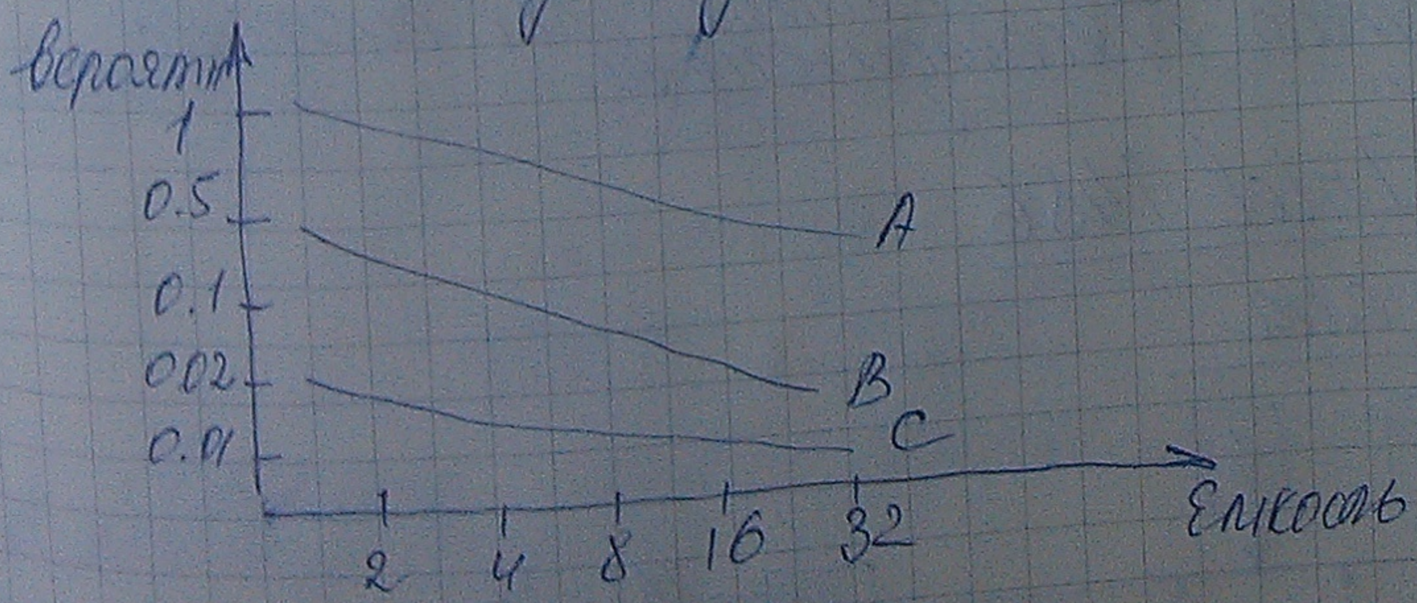
Кол-во блоков Q: $Q = \frac{2^n}{k}$, где k - кол-во слов в каждом блоке.

m - число блоков $m << Q$

Каждый блок отличается определенных тэгом или ярлыком, который говорит о том, что хранится в конкретном блоке КЭШ-памяти.

В процессе обслуживания 2 или 3 КЭШ-память представляет собой одно для хранения данных, другой для тэгов.

Эффективность КЭШ-памяти зависит от ряда факторов:

1) Емкость
2) Размера банка
3) Способа отбора от памяти на КЭШ
4) Алгоритм замещения в запол. КЭШ-памяти
5) Алгоритм согласования содерж. КЭШ и осн.
6) Число уровней КЭШ-памяти

Чем вместительней КЭШ-память, тем больше лог. схем, нужно учитывать в ее адресации, а это увеличивается вр. доступа